

Projeto Final - SCC 0241 Laboratório de Banco de Dados

Nome	Nº USP
Adriano da Silva de Carvalho	13692400
Giancarlo Malfate Caprino	12725025

Passo a passo para configurar o SQL Server e conectar com a Aplicação Java.

1. Baixar o SQL Server

(<https://learn.microsoft.com/pt-br/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>)

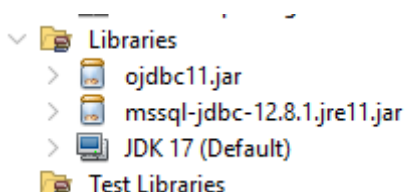
Aqui vai ser instalado o SQL Server para rodar no localhost, o nome do servidor vai ser o nome do seu computador, além disso, dois programas vão ser instalados:

- Microsoft SQL Server Management Studio 20
 - IDE para interagir com a database (sqldeveloper.exe / pgadmin)
- SQL Server Configuration Manager
 - Programa para configurar o servidor do SQL Server.

2. Baixar o Driver do JDBC para conexão com o MS SQL

(<https://learn.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?view=sql-server-ver16#82>)

Depois de baixar os drivers, tem que extrair a pasta ZIP e colocar o '.jar' nas Libraries do projeto, como estamos usando o Java 11, tem que escolher a '.jre11.jar'.

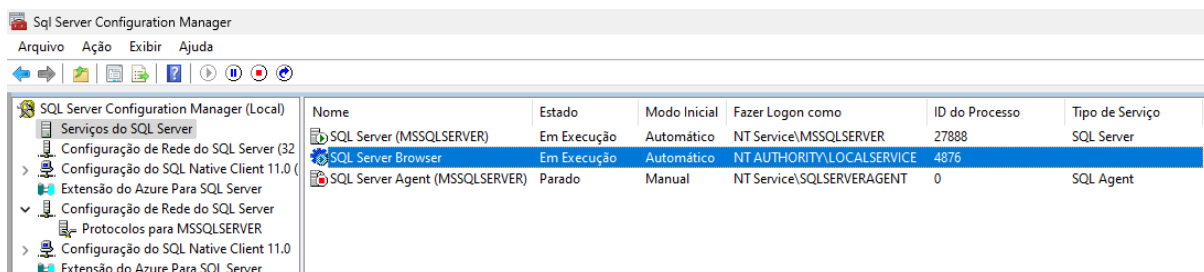


3. Configurar a conexão entre o SQL Server, a máquina local e a aplicação Java.

Aqui os principais desafios são:

3.1. Verificar os status dos serviços do SQL Server

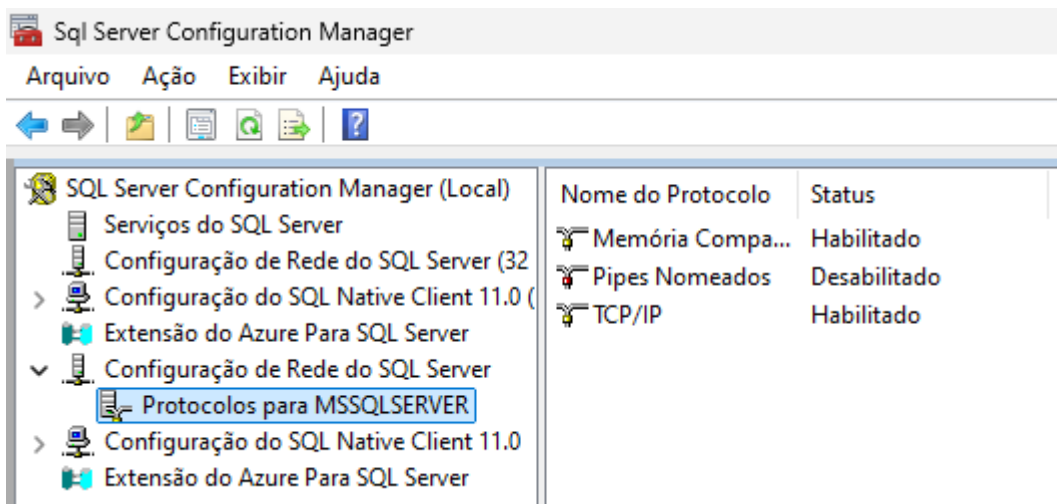
Aqui temos que ver se o *SQL Server (MSSQLSERVER)* e o *SQL Server Browser* estão em 'Execução'



Nome	Estado	Modo Inicial	Fazer Logon como	ID do Processo	Tipo de Serviço
SQL Server (MSSQLSERVER)	Em Execução	Automático	NT Service\MSSQLSERVER	27888	SQL Server
SQL Server Browser	Em Execução	Automático	NT AUTHORITY\LOCALSERVICE	4876	SQL Server
SQL Server Agent (MSSQLSERVER)	Parado	Manual	NT Service\SQLSERVERAGENT	0	SQL Agent

3.2. Autorizar conexão via TCP / IP da string de conexão da aplicação Java para o SQL Server.

Para isso, no *Sql Server Configuration Manager*, temos que garantir que as configurações de rede estão com o status 'Habilitado' para o protocolo 'TCP/IP'.



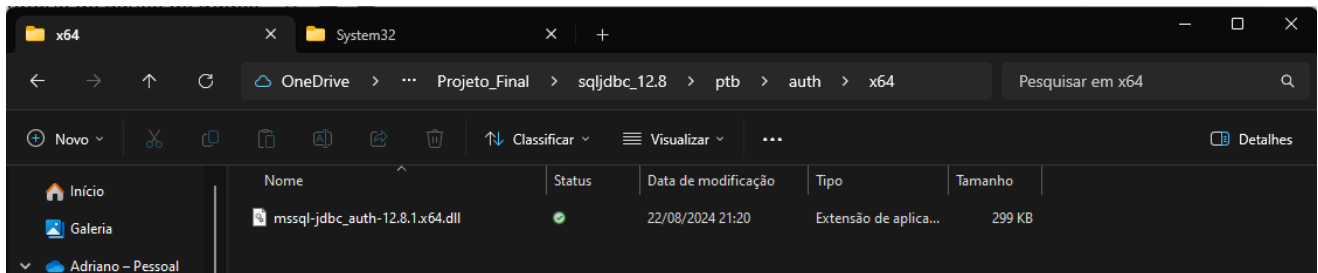
Nome do Protocolo	Status
Memória Compartilhada	Habilitado
Pipes Nomeados	Desabilitado
TCP/IP	Habilitado

3.3. Autorizar autenticação nativa, essa que é feita através do usuário do Windows (login e senha).

Para isso, vamos precisar configurar a 'dll' de autenticação.

- **Passo 1**

No driver JDBC que foi extraído, vamos **copiar** o arquivo *'mssql-jdbc_auth-12.8.1.x64.dll'* que vai estar localizado em *'sqljdbc_12.6/ptb/auth/(tipo do Sistema Operacional, no meu caso é x64)/mssql-jdbc_auth-12.8.1.x64.dll'*



- **Passo 2**

Colar a *'dll'* na raiz da *'C:\Windows\System32'*.

4. Alterar o método *'conectar()'* para realizar a conexão com o SQL Server.

O método *'conectar()'* da classe *'DBFuncionalidades.java'* deve ser alterado para que ele utilize o Driver de conexão fornecido pela biblioteca do JDBC.

Abaixo está o trecho de código atualizado do método *'conectar()'*:

```
public boolean conectar() {
    try {
        // Registro do driver JDBC do SQL Server
        DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver());

        // String de conexão para o Schema 'LAB_BD_PROJETO' e com Autenticação do Windows
        String connectionUrl =
            "jdbc:sqlserver://ADRIANO_PC;databaseName=LAB_BD_PROJETO;integratedSecurity=true;encrypt=true;trustServerCertificate=true;";

        // Estabelece a conexão
        connection = DriverManager.getConnection(connectionUrl);
        jtAreaDeStatus.setText("Conectado ao SQL Server com sucesso!");
        return true;
    } catch (SQLException ex) {
        jtAreaDeStatus.setText("Erro ao conectar ao SQL Server: " + ex.getMessage());
        System.out.println("Erro ao conectar ao SQL Server: " + ex.getMessage());
    }
    return false;
}
```